Before convergence early stopping criterion for inner LDPC code in DVB standards

Cédric Marchand, E Boutillon

Before Convergence Early Stopping Criterion for Inner LDPC Code in DVB Standards

C. Marchand and E. Boutillon

This letter presents a "before convergence" early stopping criterion for the LDPC decoder defined in the second generation of DVB standards. The idea is to stop the decoding process once the estimated number of remaining errors is below the maximum capacity correction of the outer BCH decoder used in the DVB-S2, T2 and C2 standards. Simulations show that the average number of iterations is reduced by up to 26% compared with classical early stopping criterion.

Introduction: The DVB-S2 [1], -C2, -T2 and -S2X standards (called DVB-X2 in the sequel) were ratified by the DVB committee to specify the broadcast of TV through different channels (satellite, cable and terrestrial). The coding part of these standards contains a common inner Low Density Parity Check (LDPC) encoder and an outer Bose-Chaudhuri-Hocquenghem (BCH) encoder. Similarly, the Chinese Mobile Multimedia Broadcasting (CMMB) [2] defines an LDPC encoder and an outer Reed-Solomon (RS) encoder. The use of an Early Stopping Criterion (ESC) avoids wasting time and energy on useless decoding iterations once the inner LDPC decoder has converged to a codeword, i.e., when all the parity checks are fulfilled. In [3], Kienle *et al* propose to extend the utilization of the ESC to discriminate decodable and non-decodable blocks. In [5], the Hard-Decision-Aided (HDA) method, initially designed for Turbo codes [4], was implemented for LDPC decoders. The HDA method is simple to implement but suffers from performance degradation. The key idea of this letter is to replace the ESC by a "Before Convergence" Early Stopping Criterion (BC-ESC), i.e., to stop the iterative decoding process as soon as the estimated number of remaining errors in the information bits is below or equal to the error capacity correction of the outer code.

Principle of the "Before Convergence" Stopping Criterion: The DVB-X2 standards feature a powerful coding scheme based on an outer BCH encoder and an inner LDPC encoder. This outer code is introduced to avoid error floors at low bit-error rates. One of the key features of BCH codes is the ease with which they can be decoded. Another advantage of using BCH codes is the capability to correct all patterns of t or less errors among the bits of information, with $t = 12$ for code rates r=1/4, 1/3, 2/5, 1/2, 3/5, 3/4 and 4/5; $t = 10$ for code rates 2/3 and 5/6; $t = 8$ for code rates 8/9 and 9/10. In the DVB-X2 standard codes, all redundant bits of the LDPC code have a variable node degree equal to two and the information bits have a degree greater or equal to 3. As a consequence, information bits tend to converge faster than redundant bits. If the errors are all located on the redundancy bits, a non-null syndrome can still be associated to error free information bits. Moreover, since the BCH code can correct t errors in the information bits, it is possible to stop the decoding process as soon as the number of errors e in the information bit is bound by t ($e \leq t$), with an arbitrary number of errors in the redundant part. In the sequel, the parity check matrix contains m parity checks of degree d_c and n variables. For a given Check Node (CN) c, $\mathcal{V}(c)$ represents the set of Variable Nodes (VN) connected to CN c, $\mathcal{V}/v$ represents $\mathcal{V}$ excluding VN v. The messages from check to variable and the messages from variable to check are denoted, respectively $M_{c\rightarrow v}$ and $M_{v\rightarrow c}$; the soft output of variable v is denoted $SO(v) = ln(\frac{\Pr(v=0)}{\Pr(v=1)})$.

When the number of errors is low, the number of unsatisfied checks should also be low. From this observation, a "Before Convergence" Early Stopping Criterion can be derived. After each decoding iteration, the number τ_o of Non-satisfied CN is computed by counting the number of one's in the syndrome vector $(S_o(c), c = 0 \dots m-1)$, with $S_o(c)$ computed using the sum over GF(2) as

$$S_o(c) = \bigoplus_{v' \in \mathcal{V}(c)} \text{sign}(SO(v')), \tag{1}$$

where the sign function is the hard decision associated with the soft value x: $\text{sign}(x) = 0$ if $x > 0$, 1 otherwise. When τ_o is below a threshold T_o, the decoding process is stopped. Note that if $\tau_o = 0$, the decoded message is a codeword. In a flooding scheduling architecture, an iteration consists in updating all the check nodes (thus, the computation of the S_o), then all the variable nodes. Unfortunately, the computation of the S_o values is not carried out in a layered decoding algorithm.

Syndrome computation in layered decoders: Let us recall the principle of layered decoding [6] and associated stopping criteria. One iteration consists in sequentially processing all the CNs. The update of the VNs connected to a given CN is done serially in three steps. First, the $M_{v\rightarrow c}$ are calculated:

$$M_{v\rightarrow c} = SO(v) - M_{c\rightarrow v}^{old}, \tag{2}$$

with $M_{c\rightarrow v}^{old}$ set to 0 during the first iteration and $SO(v)$ initialized with the intrinsic information. The second step is the serial $M_{c\rightarrow v}$ update. For implementation convenience, the sign and the absolute value of the message are updated separately:

$$\text{sign}(M_{c\rightarrow v}^{new}) = \bigoplus_{v' \in \mathcal{V}(c)/v} \text{sign}(M_{v'\rightarrow c}), \tag{3}$$

$$|M_{c\rightarrow v}^{new}| = f\left(\sum_{v' \in \mathcal{V}(c)/v} f(|M_{v'\rightarrow c}|)\right), \tag{4}$$

where $f(x) = \ln \tanh\left(\frac{x}{2}\right)$. The third step is the calculation of the SO^{new} value

$$SO(v)^{new} = M_{v\rightarrow c} + M_{c\rightarrow v}^{new}. \tag{5}$$

From these equations, the node processor architecture can be derived, as depicted in Fig. 1. The left adder of the architecture performs (2) while the right adder implements (5). The central part is in charge of the serial $M_{c\rightarrow v}$ updates (i.e. (3) and (4)).

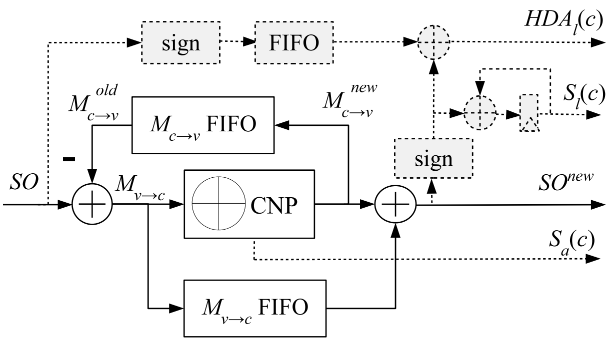

Fig. 1. Architecture of the check node processor.

The computation of τ_0 requires copying all SO values at the end of every decoding iteration and processing them independently of the layered decoding process. To avoid freezing the layered decoding process when the syndrome is computed, additional hardware is required to process the syndrome in parallel. The drawback is a more complex design and a systematic waste of one decoding iteration before the ESC is computed (with the hypothesis that the syndrome computation takes the same time as the decoding iteration). In order to overcome this problem, the syndrome can be computed on the fly during the decoding iteration. Two sub-optimal syndromes can be generated in the horizontal scheduling architecture (or layered architecture). First, the syndrome $S_l(c)$ can be computed on the fly using the current value of the soft information (5)

$$S_l(c) = \bigoplus_{v' \in \mathcal{V}(c)} \text{sign}(SO(v')^{new}). \tag{6}$$

The additional hardware required to compute $S_l(c)$ is presented in dashed lines in Fig. 1. $S_l(c)$ is set to zero when the check node c starts to be processed (this mechanism is not shown in Fig. 1). During d_c clock cycles, $S_l(c)$ accumulates the values SO^{new} to process (6). The computation of S_l will require less hardware than S_o, however S_l is sub-optimal because the SO used in (6) may change sign several times during one iteration. The classical method to compute (3) for all $v \in \mathcal{V}(c)$ is to use the fact that $M \oplus M = 0$ in GF(2), thus:

$$S_a(c) = \bigoplus_{v' \in \mathcal{V}(c)} \text{sign}(M_{v'\rightarrow c}), \tag{7}$$

$$\text{sign}(M_{c\rightarrow v}^{new}) = S_a(c) \oplus \text{sign}(M_{v\rightarrow c}). \tag{8}$$

Considering an approximate of $SO(v')$ by the variable to check message $M_{v'\rightarrow c}$ (see (2)), $S_a(c)$ can be used as an approximated version of $S_l(c)$. From $S_l(c)$ and $S_a(c)$, it is also possible to compute $\tau_l = \sum_{c=0}^{m-1} S_l(c)$ and $\tau_a = \sum_{c=0}^{m-1} S_a(c)$. Comparing τ_l with a threshold T_l and τ_a with a threshold T_a gives two new BC-ESC.

Simulation results: Bit true C simulation has been performed using the architecture presented in [7]. For each code rate, the signal to noise ratio (SNR) is set to the value required by the standard for a Quasi Error Free (QEF) transmission. For each code rate, $N = 5 \times 10^6$ frames are simulated with it_{max} iterations. A BC-ESC is assumed to be "standard compliant" if, among the N tested frames, it never stops the decoding process with more than t remaining errors. In the sequel, subscript x will denote one of the three BC-ESC methods (x = o, l or a), the average number of decoding iterations for a given BC-ESC method is noted as A_x.

Let $\tau_x(k,i)$ (respectively $e(k,i)$) be the number of non-verified syndromes (respectively remaining errors) for the k^{th} simulated codeword at iteration i ($i \le it_{max}$). Let $\mathcal{T}_x$ be the set of integers so that $T \in \mathcal{T}_x$ implies that, for all $k = 1 \ldots N-1, i = 1 \ldots it_{max}$,

$$\{\tau_x(k,i) \le T\} \Rightarrow \{e(k,i) \le t\}. \tag{9}$$

The threshold T_x is thus determined as $T_x = \max\{\mathcal{T}_x\}$. A two dimensional BC-ESC implying both τ_l and τ_a can be defined by extending (9) to the two dimensional case with a dual-criteria BC-ESC. Let $\mathcal{T}_{l,a}$ be the set of couples $(\hat{T}_l, \hat{T}_a) \in \mathbb{N}^2$ so that, for all $k = 1 \ldots N-1, i = 1 \ldots it_{max}$:

$$\{\tau_l(k,i) \le \hat{T}_l, \tau_a(k,i) \le \hat{T}_a\} \Rightarrow \{e(k,i) \le t\}. \tag{10}$$

At a given iteration, if $(\hat{T}_l = \tau_l(k,i), \hat{T}_a = \tau_a(k,i))$ belongs to $\mathcal{T}_{l,a}$, then, according to (10), the number of residual errors is below t, thus the LDPC decoder can stop the iterative process and outputs the current hard decision to the outer BCH code. Fig. 2 represents the set $\mathcal{T}_{l,a}^{4.68}$ for a code rate $r = 4/5$ and a SNR of 4.68 dB (SNR given by the DVB-S2 standard specification for QEF decoding). $\mathcal{T}_{l,a}^{4.68}$ is defined as the set of points below the curve $\bar{T}_a^{4.68}$. Testing whether a point belongs to this set or not is straightforward: for a given value of τ_l the maximum admissible value $\bar{T}_a^{4.68}(\tau_l)$ is retrieved from a memory and τ_a is compared with $\bar{T}_a^{4.68}(\tau_l)$. For example, if $\tau_l = 3$ then $\bar{T}_a^{4.68}(3) = 100$, thus if $\tau_l = 3$ and $\tau_a \le 100$ then the decoding process stops. Fig. 2 also shows the sets $\mathcal{T}_{l,a}^{4.80}$ and $\mathcal{T}_{l,a}^{4.90}$ (for SNR 4.80 dB and 4.90 dB respectively). One can note that $\mathcal{T}_{l,a}^{4.68} \subset \mathcal{T}_{l,a}^{4.80} \subset \mathcal{T}_{l,a}^{4.90}$, thus, using the set $\mathcal{T}_{l,a}^{4.68}$ for the BC-ESC, no errors are introduced for SNR 4.8 dB and 4.9 dB. This property can be generalized for all SNR above 4.68 dB and all code rates.

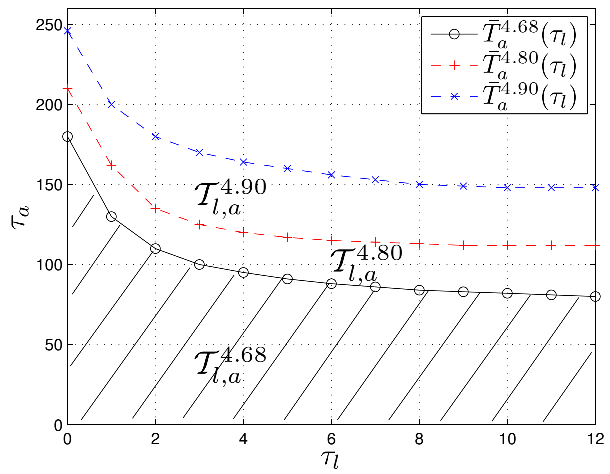

Fig. 2 *Sets $\mathcal{T}_a^{4.68}$, $\mathcal{T}_a^{4.80}$ and $\mathcal{T}_a^{4.90}$ for code rate $r = 4/5$. The limits of these sets are defined by the x-axis, the y-axis and the corresponding curve $\bar{T}_a$* . $\mathcal{T}_a^{4.68}$ is represented in dashed lines.

Table 1 shows T_x and A_x for different rates. By convention $A_x(T_x)$ indicates the average number of iterations when the BC-ESC $\tau_x \le T_x$ is used. A_g is the minimum average number of iterations obtainable when a genius BC-ESC stops the decoding process as soon as $e \le t$. Considering the state of the art ESC A_o, one iteration is added to take into account the latency for the syndrome computation processing in a layered architecture. Note that, in [6], convergence is detected when all $HDA_l(c) = 0$ (no change of SO sign during two consecutive iterations, see Fig. 1) and when $\tau_a = 0$ (all check nodes are fulfilled). This ESC also requires an extra decoding iteration after convergence, again leading to an average number of decoding iterations equal to $A_o + 1$.

For all code rates, A_l, A_a and $A_{l,a}$ are below $A_o + 1$, except for code rates $r = 2/3$, $3/4$ and $4/5$ where $A_l(T_l) = it_{max}$. For these code rates, $T_l = -1$, which implies that condition (9) is never fulfilled. In fact, as shown on Fig. 2, when $\tau_l = 0$, if $\tau_a > 180$ then e can be greater than t, i.e $\tau_l = 0$ is not a sufficient criterion to ensure that $e \le t$. The last line of Table 1 shows the reduction gain between the classical ESC ($A_o + 1$) and the proposed BC-ESC when the dual-criteria $A_{l,a}$ is used. The gain varies from 8 % (high code rates) up to 26% for code rate 1/4. Performance of the genius BC-ESC shows that there is still a significant potential reduction in the average number of decoding iterations for low code rates (for code rate $r = 1/4$, $A_g = 15.7$ while $A_{l,a} = 20.1$).

Combined with an input buffer, the BC-ESC can be used to increase the decoding throughput [8]. It can also be used to reduce the average energy required to decode a codeword.

Table 1: Average number of iterations for several ESC methods

code rate r	1/4	1/3	2/5	1/2	3/5	2/3	3/4	4/5
it_{max}	45	40	35	35	30	30	30	30
SNR (dB)	-2.35	-1.24	-0.3	1	2.35	3.1	4.03	4.68
A_g	15.7	17.3	15	13.5	8.8	11.7	11.1	9.5
$A_o + 1$	27.2	25.5	20.5	18.2	12.1	15.4	13.6	11.9
T_l	0	0	0	0	0	-1	-1	-1
$A_l(T_l)$	26.1	24.5	19.6	17.4	11.2	30	30	30
T_a	309	281	384	274	274	79	141	108
$A_a(T_a)$	23	22.6	18.5	16.1	11	14.3	12.6	11.2
$A_{l,a}$	20.1	20.6	17.5	15.3	10	13.5	12.5	10.9
gain (in %)	26.1	19.2	14.6	15.9	17.	12.3	8.1	8.4

Conclusion: This letter shows that stopping the decoding process of an inner LDPC code can significantly reduce the average number of decoding iterations without any performance degradation, when the estimated number of remaining errors is below the maximum capacity correction of the outer decoder. A low complexity dual-criteria BC-ESC has also been proposed for DVB-S2, T2 and C2 standards. Compared to ESC, BC-ESC can reduce the average number of iterations of a few percents up to 26 %, depending on the code rate. The proposed BC-ESC can be used for energy saving or for increasing the average decoding throughput.

C. Marchand and E. Boutillon (*Lab-STICC, UMR 6285, Université de Bretagne Sud, France*)

E-mail: {emmanuel.boutillon}@univ-ubs.fr

References

1 Digital Video Broadcasting (DVB): ‘Second generation framing structure, channel coding and modulation systems for Broadcasting, Interactive Services, News Gathering and other broadband satellite applications (DVB-S2)’, ETSI EN 302 307, V1.2.1, April 2009

2 GY/T 220.1-2006: ‘Mobile multimedia broadcasting part 1: framing structure, channel coding and modulation for broadcasting channel’, 2006.

3 Kienle, F. and Wehn, N.: ‘Low complexity stopping criterion for LDPC code decoders’, *Vehicular Technology Conference*, 2005, **1**, pp.606-609

4 Shao, R.Y., Lin, S. Fossorier, M.P.C.: ‘Two simple stopping criteria for turbo decoding’, *IEEE Trans. Communications*, 1999 , **47**, (8), pp.1117-1120

5 Shih, X.Y., Zhan, C.Z., Lin, C.H., Wu, A.Y.: ‘An 8.29 mm^2 52 mW Multi-Mode LDPC Decoder Design for Mobile WiMAX System in 0.13 μm CMOS Process’, *IEEE Journal of Solid-State Circuits*, 2008, **43**, (3), pp.672-683

6 Hocevar, D.E.: ‘A reduced complexity decoder architecture via layered decoding of LDPC codes’, *IEEE Workshop on Signal Processing Systems (SiPS)*, 2004, pp.107-112

7 Marchand, C., Conde-Canencia, L., Boutillon, E.: ‘High-speed conflict-free layered LDPC decoder for the DVB-S2, -T2 and -C2 standards’, *IEEE Workshop on Signal Processing Systems (SiPS)*, 2013, pp.118-123

8 Bosco G., Montorsi G., Benedetto S.: ‘Decreasing the complexity of LDPC iterative decoders’, *IEEE Communications Letters*, 2005, **9**, (7), pp.634-636.